Informe de Seguridad Detallado

URL Analizada: rosenbergcanada.com
IP del Sitio: 107.180.41.37
Fecha del Análisis: 30 de septiembre de 2024
Herramienta Utilizada: OWASP ZAP
Versión de PHP Detectada: 7.3.23
Calificación General de Seguridad: F (Riesgo Alto)

Resumen del Análisis de Seguridad

El análisis de seguridad del sitio web `rosenbergcanada.com` reveló diversas vulnerabilidades críticas que requieren atención inmediata. Estas debilidades comprometen la integridad del sitio y la privacidad de los usuarios, lo que podría llevar a ataques como **Cross-Site Scripting (XSS)**, **Clickjacking** y **exposición de información sensible**. Además, la falta de políticas y configuraciones de seguridad estándar agrava estos riesgos.

A continuación se presentan los detalles de las vulnerabilidades encontradas y las recomendaciones específicas para mejorar la seguridad del sitio.

Vulnerabilidades Críticas Detectadas

1. Cabeceras de Seguridad Faltantes

La ausencia de cabeceras de seguridad críticas expone el sitio a diferentes tipos de ataques. Las cabeceras de seguridad son configuraciones que permiten a los navegadores proteger mejor las interacciones con el sitio.

- **Strict-Transport-Security (HSTS)**:
  - **Problema**: No se ha configurado el encabezado **HSTS**, lo que significa que el sitio no está forzando el uso de HTTPS. Esto permite que las conexiones no seguras (HTTP) puedan ser interceptadas en un ataque "man-in-the-middle".

    - **Recomendación**: Implementar **Strict-Transport-Security** con el valor `max-age=31536000; includeSubDomains` para asegurar que el sitio utilice exclusivamente HTTPS y proteja las comunicaciones con los usuarios.
- **Content-Security-Policy (CSP)**:
    - **Problema**: El sitio carece de una política de seguridad de contenido, lo que lo deja vulnerable a ataques de inyección de scripts maliciosos (XSS).
    - **Recomendación**: Configurar un **Content-Security-Policy** adecuado que limite los orígenes de contenido permitidos. Por ejemplo, solo permitir contenido de fuentes confiables (`default-src 'self'`).
- **X-Frame-Options**:
    - **Problema**: La falta de esta cabecera expone al sitio a ataques de **clickjacking**, donde los atacantes pueden incrustar la página en un iframe y engañar al usuario para que haga clic en un enlace malicioso.
    - **Recomendación**: Agregar la cabecera **X-Frame-Options** con el valor `SAMEORIGIN` para prevenir que el sitio sea embebido en iframes de otros dominios no autorizados.
- **X-Content-Type-Options**:
    - **Problema**: Sin la cabecera **X-Content-Type-Options**, el sitio permite a los navegadores realizar "MIME-type sniffing", lo que podría llevar a la ejecución de archivos peligrosos en lugar de tratarlos como archivos seguros.
    - **Recomendación**: Implementar la cabecera **X-Content-Type-Options** con el valor `nosniff` para evitar que el navegador interprete incorrectamente los tipos MIME.
- **Referrer-Policy**:
    - **Problema**: La ausencia de una política de referencia puede provocar que URLs sensibles, que contienen información privada, sean enviadas a sitios externos cuando los usuarios hacen clic en un enlace.
    - **Recomendación**: Agregar una **Referrer-Policy** para controlar la información que se comparte al salir del sitio. Un valor recomendado podría ser `strict-origin-when-cross-origin` .
- **Permissions-Policy**:
    - **Problema**: Sin la cabecera **Permissions-Policy**, el sitio no limita adecuadamente el acceso a funciones sensibles del navegador, como la cámara, micrófono o geolocalización.
    - **Recomendación**: Implementar la cabecera **Permissions-Policy** para restringir el uso de estas características solo cuando sea necesario.

2. Exposición de Información Sensible

- **Server Banner Visible**:
  - **Problema**: El servidor revela la versión de **PHP (7.3.23)** y **Apache**, lo que facilita a los atacantes dirigirse a vulnerabilidades conocidas para esas versiones.
  - **Recomendación**: Ocultar la versión del servidor mediante la configuración de los encabezados `Server` y `X-Powered-By` para que no se muestre información sobre el software utilizado. Además, se recomienda actualizar PHP a la versión más reciente, ya que **PHP 7.3** ha alcanzado su fin de vida útil y ya no recibe parches de seguridad.

3. Inyección de Scripts (XSS)

- **Cross-Site Scripting (XSS)**:
  - **Problema**: Se encontraron varias entradas en el sitio que no tienen una validación o sanitización adecuada. Esto permite a los atacantes inyectar código malicioso que podría ejecutarse en el navegador de un usuario desprevenido.
  - **Recomendación**: Implementar sanitización en todas las entradas del usuario y usar mecanismos de escape para prevenir la ejecución de código no autorizado en el navegador.

4. Cookies Inseguras

- **Falta de Flags de Seguridad en las Cookies**:
  - **Problema**: Las cookies del sitio no tienen configurados los flags de seguridad **Secure**, **HttpOnly**, y **SameSite**, lo que las hace vulnerables a ser interceptadas o modificadas en ataques.
  - **Recomendación**: Asegurarse de que todas las cookies sensibles se configuren con los siguientes atributos:
    - **Secure**: Solo permite que la cookie se transmita a través de conexiones HTTPS.
    - **HttpOnly**: Evita el acceso a las cookies mediante JavaScript, protegiendo los datos de sesión.
    - **SameSite**: Controla cómo las cookies se envían en solicitudes de terceros, protegiendo contra ataques CSRF.

Recomendaciones Generales

1. **Mejorar la Seguridad de la Conexión**:
   - Habilitar **Strict-Transport-Security (HSTS)** para garantizar que todas las conexiones al sitio se realicen a través de HTTPS.
2. **Implementar Políticas de Seguridad de Contenidos (CSP)**:
   - Limitar las fuentes de scripts y otros recursos para mitigar el riesgo de inyección de contenido malicioso.
3. **Prevenir Ataques de Clickjacking**:
   - Usar **X-Frame-Options** para evitar que el sitio sea embebido en iframes no autorizados.
4. **Ocultar Información del Servidor**:
   - Configurar el servidor para no mostrar la versión de **PHP** y otros detalles del servidor que podrían ser utilizados en ataques dirigidos.
5. **Actualizar la Versión de PHP**:
   - Migrar a una versión más reciente de PHP (por lo menos PHP 8.x) que siga recibiendo soporte y actualizaciones de seguridad.
6. **Configurar Correctamente las Cookies**:
   - Asegurarse de que las cookies sensibles tengan los flags **Secure**, **HttpOnly**, y **SameSite** para proteger la privacidad del usuario.

Conclusión

El sitio web **rosenbergcanada.com** presenta varias vulnerabilidades de seguridad que deben ser abordadas de inmediato. Los problemas más críticos están relacionados con la falta de cabeceras de seguridad esenciales, la exposición de información del servidor y la configuración insegura de las cookies. Solucionar estos problemas mejorará significativamente la seguridad del sitio y reducirá el riesgo de ataques cibernéticos.

Se recomienda seguir las medidas correctivas proporcionadas para cada vulnerabilidad, además de mantener el sitio y sus componentes actualizados regularmente para garantizar una seguridad continua.